**Examen Parcial**

Carrera **:**  ING. INFORMÁTICA

Materia **:** PD

Alumno : Turno : Div.: “ “

Tema**: Haskell Tema I** Legajo: Fecha :

PUNTAJE: \_\_\_\_\_\_\_\_\_\_\_\_\_ CALIFICACIÓN DEFINITIVA: \_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_

**OLARIAGA OLIVETO, JESÚS**

**Criterios:**

Escribir el código

Pegar un print de la ejecución

**Desarrollo**

1. Definir la función conc\_lista tal que conc l1 l2 es la concatenación de l1 y l2.

Por ejemplo, conc\_lista [2,3] [3,2,4,1] -> [2,3,3,2,4,1]

Nota: conc\_lista es equivalente a (++).

conc\_lista :: [a] -> [a] -> [a]

conc\_lista l1 l2 = foldr (:) l2 l1

1. Redefinir la función init tal que init\_list l es la lista l sin el último elemento.

Por ejemplo, init\_list [1,2,3] -> [1,2]

Init\_list [4] -> []

init\_list :: [a] -> [a]

init\_list l = init l

1. Redefinir la función length tal que length\_list l es el número de elementos de l.

Por ejemplo, length\_list [1,3,6] -> 3

length\_list :: [a] -> Int

length\_list l = length l

1. Definir la función palíndromo tal que (palíndromo xs) se -- verifica si xs es un palíndromo; es decir, es lo mismo leer xs de -- izquierda a derecha que de derecha a izquierda.

Por ejemplo, palindromo [3,2,5,2,3] -> True

palindromo [3,2,5,6,2,3] -> False

palindromo :: Eq a => [a] -> Bool

palindromo l = l == reverse l

1. Implementar uma función que devuelva una lista con los n elementos mayores de la lista original.

Por ejemplo, list\_may 3 [1,3,4,2,6] -> [3,4,6]

list\_may 2 [1.3.4.2.6] -> [4,6]

list\_may :: Ord a => Int -> [a] -> [a]

list\_may n l = take n (reverse (sort l))

PRUEBAS (EN ORDEN)

